|  |  |
| --- | --- |
| **Đại HỌC QUỐC GIA HÀ NỘI** | |
| **tRƯỜNG đẠI HỌC CÔNG NGHỆ** | |
| **----------------------------------------** | |
|  | |
| BÁO CÁO BÀI TẬP LỚN | |
| NHÓM 1 | |
|  | |
| **ĐỀ BÀI: TÌM HIỂU PHANTOMJS** | |
|  | |
| Giảng viên: Trương Anh Hoàng | |
|  | |
|  | **Hoàng Kim Băng** |
|  | **Hoàng Đức Tùng** |
|  | **Nguyễn Việt Hoàng** |
|  | **Hà Thanh Loan** |
|  |  |
|  |  |
| **Hà Nội, tháng 10 năm 2013** | |

MỤC LỤC

Mục lục

[I. Tìm hiểu chung 2](#_Toc369951664)

[1. Tổng quan về PhantomJS 2](#_Toc369951665)

[2. Lịch sử phát triển của PhantomJS 2](#_Toc369951666)

[3. PhantomJS và NodeJS. 3](#_Toc369951667)

[II. Chức năng chính của PhantomJS. 4](#_Toc369951668)

[1. Headless Website Testing 4](#_Toc369951669)

[a. Headless Testing 4](#_Toc369951670)

[b. Không là Test FrameWork 4](#_Toc369951671)

[c. Các dự án 5](#_Toc369951672)

[d. Hệ thống tích hợp liên tục 6](#_Toc369951673)

[2. Screen Capture 6](#_Toc369951674)

[3. Page Automation 9](#_Toc369951675)

[a. Thao tác trên DOM 9](#_Toc369951676)

[b. Sử dụng jQuery và các thư viện khác 10](#_Toc369951677)

[4. Network monitoring 10](#_Toc369951678)

[III. Bắt đầu với PhanTomJS. 11](#_Toc369951679)

[1. Cài đặt 11](#_Toc369951680)

[2. Ví dụ 14](#_Toc369951681)

[IV. Ứng dụng thực tế của PhantomJS 17](#_Toc369951682)

1.YSlow………………………………………………………………………………………………….17

2.QUnit…………………………………………………………………………………………………..20

V. Tổng kết…………………………………………………………………………………………..21

Tài liệu tham khảo………………………………………………………………………………………...21

**TÌM HIỂU PHANTOMJS**

# Tìm hiểu chung

## Tổng quan về PhantomJS

Trước khi hiểu PhantomJS là gì thì chúng ta sẽ phải hiểu Webkit là gì? Có thể hiểu Webkit là một khung ứng dụng cung cấp đầy đủ các thành phần để lập trình nên một trình duyệt Web. Có nghĩa là nền tảng để tạo nên các trình duyệt như FireFox, Chrom, Safari … chính là Webkit.

Vậy PhantomJS là gì? Nó là một trình duyệt Web nhưng là một trình duyệt không đầu. Có nghĩa là các kết quả được trả về trên Web sẽ không được hiển thị rõ ràng mà PhantomJS sẽ cung cấp các kịch bản, cú pháp để người dùng thao tác trực tiếp với Webkit và trả về kết quả mong muốn.

Ngôn ngữ tạo lên PhantomJS là JavaScript một ngôn ngữ thực sự quen thuộc với mỗi nhà lập trình.

PhantomJS là một mã nguồn mở, nó hỗ trợ các chuẩn Web: DOM handling, CSS selector, JSON, Canvas và SVG.

## Lịch sử phát triển của PhantomJS

PhantomJS là một dự án mã nguồn mở của Ariya Hidayat, một người sinh ra tại Indonesia. Cùng với Esprima, PhantomJS là một trong hai dự án lớn nhất của ông. Ông có trên 30 bài báo, bằng tiến sĩ tại đại học Paderborn của Đức.

Đến nay PhantomJS vẫn không ngừng phát triển và có đã trải qua khoảng chín phiên bản.

Phiên bản chính thức đầu tiên là PhantomJS 1.1 phát hành vào ngày 27 tháng 4 năm 2011 và được Ariya Hidayat đặt tên là Cherry Blossom. Có một điều khá thú vị của Ariya Hidayat đó là mỗi phiên bản phát hành của ông đều được đặt một tên riêng và chúng đều có mang ý nghĩa riêng. Ví dụ như cái tên Cherry Blossom, đó là tên một loại hoa nở vào mùa xuân của đất nước Nhật Bản, ông gửi đến sự trân trọng với người Nhật và muốn PhantomJS phát triển mạnh mẽ như mùa xuân.

Phiên bản mới nhất đến nay là PhantomJS 1.9.2 được phát hành vào ngày 20 tháng 3 năm 2013, có thể thấy PhantomJS vẫn luôn được phát triển với các phiên bản được tung ra khá đều đặn.

Mỗi phiên bản mới của PhantomJS là sự nâng cấp của phiên bản trước. Nâng cấp ở đây là sửa lỗi và thêm tính năng mới. Để hiểu rõ sự nâng cấp nay chúng ta sẽ tìm hiểu về sự thay đổi của phiên bản 1.9 so với 1.8. Đã có rất nhiều tính năng và cải tiến mới của bản PhantomJS 1.9 mà có thể kể đến là:

* Thêm khả năng hủy bỏ yêu cầu mạng
* Thêm chức năng tìm kiếm các dòng
* Thêm hỗ trợ mã hóa tùy chỉnh
* Thêm chỉ số tải trang và tiến độ
* Cố định xây dựng trên FreeBSD
* Sửa lỗi lưu sai ngày trong các tệp tin cookie

Trên đây chỉ là một số tính năng mới của bản PhantomJS 1.9. Bản 1.9 cũng sẽ là bản cuối cùng trong chuỗi 1.x với hứa hẹn các phiên bản 2.x mang đến tính đột phá mới về nền tảng cơ bản hiện đại hơn cùng một module Webkit cập nhật.

## PhantomJS và NodeJS.

NodeJS là một khung mã nguồn mở JavaScript cung cấp các tiện ích để phát triển các ứng dụng trên nền Web. NodeJS ra đời trước PhantomJS và có một vài đặc điểm không thể bỏ qua như: ngôn ngữ tiện dụng, phổ biến; tốc độ nhanh (nhờ V8); nhiều thư viện, dễ cài; chia thành các module để dễ quản lý…

Sau khi phát hành PhantomJS thì ngay lập tức có một câu hỏi được đặt ra: Tại sao không xây dựng PhantomJS thành một module của NodeJS, điều này sẽ giúp cải thiện hệ thống (vì nếu sử dụng cả hai thì sẽ phải cài đặt cả nền tảng PhantomJS và NodeJS) cũng như dễ quản lý hơn với người dùng?

Nền tảng của vấn đề rất hợp lý. Thứ nhất, PhantomJS và NodeJS đều là khung ứng dụng cho Website. Thứ hai chúng đều sử dụng JavaScript. Thứ ba PhantomJS ít phổ biến hơn so với NodeJS. Nhưng tại sao lại không xây dựng PhantomJS thành một module của NodeJS?

Có thể lý giải vấn đề này như sau: đầu tiên NodeJS không hỗ trợ các chức năng cần thiết khi xây dựng PhantomJS. Thứ hai, các module trong Node để chạy được thì sẽ phải nạp vào “server Node” mà trong khi đó PhantomJS cần phải có quyền kiểm soát cho mọi thứ như: vòng lặp sự kiện, lưu lượng mạng khả năng biên dịch mã JavaScript,…

Trên đây là sự khác biệt cơ bản giữa NodeJS và PhantomJS.

# Chức năng chính của PhantomJS.

## Headless Website Testing

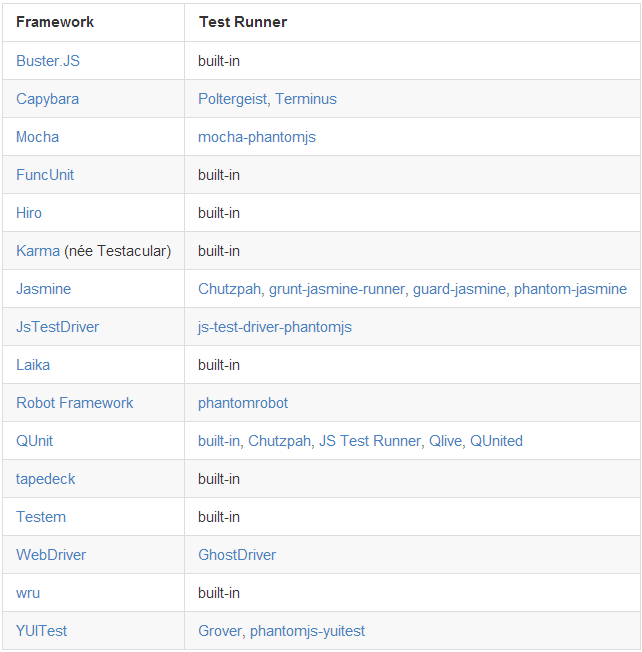
### Headless Testing

Chức năng đầu tiên của PhantomJS là kiểm thử không đầu (Headless Testing). Nó phù hợp để kiểm thử từng dong lệnh và một phần của hệ thống tích hợp liên tục.

### Không là Test FrameWork

PhantomJS không phải là một test framework, nó chỉ được sử dụng để khởi tạo các cuộc kiểm tra thông qua các test thử nghiệm phù hợp.

Bảng tóm tắt dưới đây là danh sách các test framework khác nhau và chạythử tương ứng. Nếu framework không cần sự thiết yếu hoặc “third- party runner”, nó được đánh dấu là: “tích hợp” (built-in).



Hình ảnh 01

### Các dự án

Có một số dự án được xây dựng trên nền PhantomJS để cung cấp các chức năng thuận tiện cho quá trình kiểm thử:

* CasperJS: phù hợp cho việc xây dựng các kịch bản chuyển hưởng và kiểm thử.
* Lotte: thêm phương thức jQuery-like, chuỗi và nhiều khẳng định logic.
* WebSpecter là một BDD-style (behavior-driven development) cho phép kiểm thử framework của ứng dụng Web.

### Hệ thống tích hợp liên tục

PhantomJS có thể cài đặt với hệ thống CI (Continuous Integration) . Một hệ thống tổ chức CI nổi tiếng có tích hợp sẵn PhantomJS là Travis CI.

PhantomJS cũng có thể tích hợp trên Linux.

## Screen Capture

Webkit là một phương tiện layout và reandering thực, vì vậy PhantomJS có khả năng chụp ảnh trang Web như chụp ảnh màn hình. Các nội dung được chụp có thể là trang Wev tĩnh hoặc được truy xuất từ HTML, CSS, SVG và Canvas.

Đoạn mã dưới đây sẽ chụp lại ảnh một trang Web đơn giản. Nó tải trang chủ của Github và lưu lại thành tập tin ảnh: giphub.png.

*var page = require(‘webpage’).create();*

*page.open(‘*[*http://github.com/*](http://github.com/)*’, function() {*

*page.render(‘github.png’);*

*phantom.exit();*

*});*

Hình ảnh mã nguồn 01.

* *Cách chạy đoạn mã trên sẽ được trình bày chi tiết ở phần sau.*

Ở ví dụ trên định dạng ảnh được sử dụng là đuôi .PNG, ngoài định dạng này ra PhantomJS cũng hỗ trợ thêm các định dạng khác nữa như: GIF, JPEG và PDF.

Để chứng minh khả năng dựng hình hoàn chỉnh của PhantomJS thì ta sẽ xét thêm một vài ví dụ:

Ví dụ đầu tiên là đoạn mã [rasterize.js](https://github.com/ariya/phantomjs/blob/master/examples/rasterize.js) trong thư mục example, nó sẽ truy vấn SVG và dựng hình con hổ khi được biên dịch:

*Phantomjs rasterize.js* [*http://ariya.github.com/svg/tiger.svg*](http://ariya.github.com/svg/tiger.svg) *tiger.png*

Hình ảnh mã nguồn 02.

Kết quả được hiển thị là ảnh con hổ trong bức ảnh *tiger.png:*



Hình ảnh 02.

Ví dụ tiếp theo là truy vấn HTML vẫn qua đoạn mã nguồn [rasterize.js](https://github.com/ariya/phantomjs/blob/master/examples/rasterize.js). Đoạn mã biên dịch:

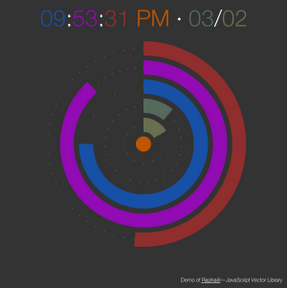
*Phantomjs rasterize.js* [*http://raphaelis.com/polar-clock.html*](http://raphaelis.com/polar-clock.html) *clock.png*

http**:***//raphaeljs.com/polar-clock.html clock.png* http**:***//raphaeljs.com/polar-clock.html clock.png*

http**:***//raphaeljs.com/polar-clock.html clock.png*

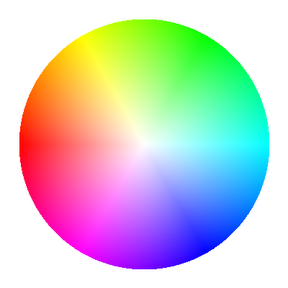
Hình ảnh mã nguồn 03.

Và đây là kết quả thu được:



Hình ảnh 03.

Ví dụ cuối cùng sẽ là về Canvas. Canvas cũng có thể dễ dàng truy xuất và xây dựng thành hình ảnh. Đoạn mã [colorwheel.js](https://github.com/ariya/phantomjs/blob/master/examples/colorwheel.js) sẽ tạo ra bánh xe màu dưới đây:



Hình ảnh 04.

Qua các ví dụ trên chúng ta có thể thấy khả năng dựng hình từ các trang Web của PhantomJS. Đã có vài dự án dựa vào PhantomJS để hỗ trợ dựng hình như: [Screener](http://screener.brachium-system.net/) và [ChromaNope.](http://chromanope.com/)

## Page Automation

PhantomJS có thể tải và thao tác trên một trang Web như một giao diện duyệt Web bình thường. Nhờ vậy PhantomJS có thể thực hiện các thao tác tự động hóa khác nhau.

### Thao tác trên DOM

### Khi các dữ liệu được hiển thị trên các trình duyệt Web cũng có nghĩa là dữ liệu từ DOM hay CSS đã hoạt động tốt.

### Ví dụ sau đây sẽ in ra thuộc tính của phần tử có id là myagent khi được biên dịch. Đoạn mã useragent.js.

### 

*var page=require(‘webpage’).create();*

*console.log(‘The default user agent is’ + page.settings.userAgent);*

*page.settings.userAgent=’SpecialAgent’;*

*page.open(‘http://www.httpuseragent.org’, function(status){*

*if(status!=’success’){*

*console.log('Unable to access network');*

*}* ***else*** *{*

***var*** *ua* ***=*** *page.evaluate(****function*** *() {*

***return*** *document.getElementById('myagent').textContent;*

*});*

*console.log(ua);*

*}*

*Phantom.exit();*

*});*

### Hình ảnh mã nguồn 04.

### Ví dụ trên cho thấy khả năng tùy chỉnh các nhân viên (user agent) từ xa.

### Sử dụng jQuery và các thư viện khác

### Sử dụng page.includeJs để thao tác với jQuery

*var page=require(‘webpage’).create();*

*page.open(‘http://www.sample.com’, function(){*

*page.includeJs(*[*http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js*](http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js)*, function(){*

*page.evaluate(function(){*

*$(“button”).click();*

*}):*

*Phantomjs.exit();*

*});*

*});*

Hình ảnh mã nguồn 05.

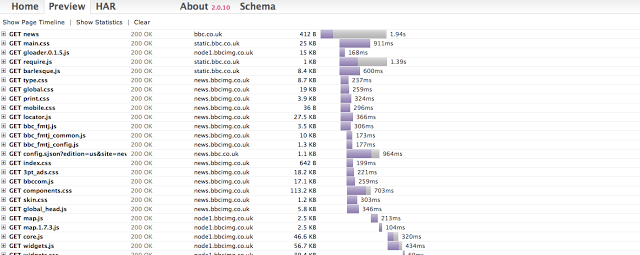
Đoạn mã trên sẽ tải trang Web [*www.sample.com*](http://www.sample.com)gồm có các jQuery và bấm vào tất cả các nút có sử dụng jQuery. Sau đó nó sẽ thoát khỏi trang Web.

## Network monitoring

PhantomJs cho phép kiểm tra lưu mạng, từ đó xây dựng và phân tích các hành vi và hiệu suất mạng.

Sử dụng *onResourceRequested* và *onResourceReceived* để tải các tài nguyên và thực hiện phân tích lưu lượng mạng.

Sau đây là một sơ đồ thác nước thu được từ trang Web BBC:



Hình ảnh 05.

Các hai dự án liên quan đến PhantomJS giúp phân tích mạng hiệu quả hơn là: Confess.js và YSlow. Trong đó Yslow rất hữu ích cho việc giám sát hiệu năng Web tự động.

# Bắt đầu với PhantomJS.

## Cài đặt

Như bao nhiêu ứng dụng khác, để sử dụng được PhantomJS thì chúng ta cũng phải cần cài đặt biến môi trường của nó trong máy tính của mình.

Sau đây là các bước cài đặt cho PhantomJS:

Bước 1: Lên trang chủ của PhantomJS (phantomjs.org) tải bản mới nhất về. Chú ý tải bản tương ứng với hệ điều hành của máy tính.

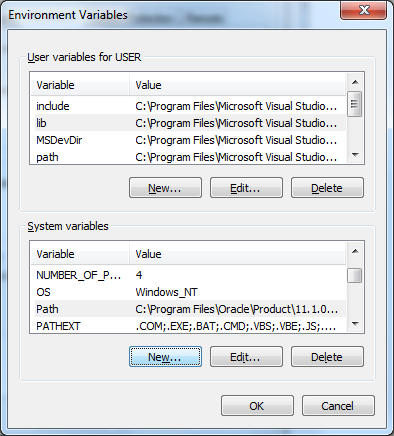
Phiên bản dùng ở ví dụ này là bản PhantomJS 1.9.2 và dùng cho Window.

Bước 2: Cài đặt.

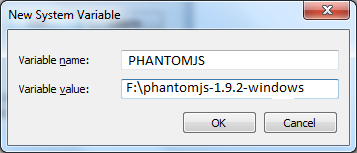
Ngoài màn hình máy tính chúng ta bấm chuộc phải vào biểu tượng **Computer/ chọn Properties.**

Một cửa sổ mới hiện lên, chúng ta lần lượt đi đến các đường dẫn sau: **Advanced System Settings** -> chọn tab Advanced -> chọn **Environment Variables.**

Trong cửa sổ **Environment variables** chọn mục Sy**stem variables rồi** chọn button .**New**.

  
Hình ảnh 06.

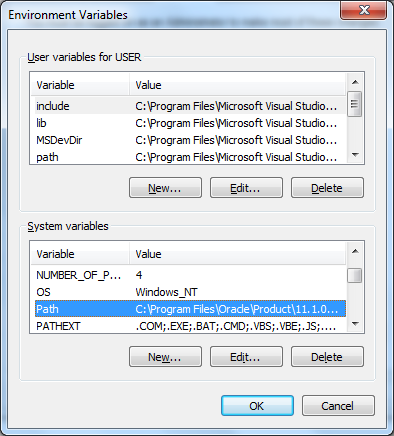
Một bảng mới hiện nên ta lần lượt điền các thông tin như hình vẽ sau:



Hình ảnh 07.

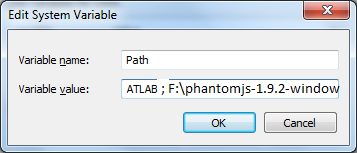
Sau khi điền xong thông tin, click chọn OK.

Trong mục System Variables tìm tới biến Path rồi chọn mục Edit.



Hình ảnh 08.

Trong mục **Variable value**, các bạn di chuyển tới cuối, copy thư mục phantomjs vào:



Hình ảnh 09.

Click chọn OK là đã hoàn thành quá trình cài đặt PhantomJS.

Bước 3. Biên dịch chương trình

Tạo một file mã nguồn JavaScript rồi lưu đoạn mã nguồn sau vào:

*Console.log(‘Hello!World’);*

*Phantomjs.exit();*

Hình ảnh mã nguồn 06.

Giả sử chương trình được lưu với tên file là *hello.js* và có đường dẫn là *F:\JS\CodeJS\hello.js* .

Mở command line trong máy tính rồi gõ lệnh:

***phantomjs F:\JS\CodeJS\hello.js***

Chương trình sẽ in ra *Hello!World.*

Đến đây đã hoàn thành quá trình biên dịch một file JavaScript bằng PhantomJS.

## Ví dụ

Sau đây là một vài ví dụ quan trọng được biên dịch bằng PhantomJS để thể hiện rõ các chức năng cũng như đặc tính riêng của ứng dụng PhantomJS

### zing.js

*console.log("Start");*

*var page = require('webpage').create();*

*page.onConsoleMessage = function(msg) {*

*console.log(msg);*

*};*

*page.open("http://login.me.zing.vn/", function(status) {*

*page.render('login\_zing\_s1.png');*

*if ( status === "success" ) {*

*page.evaluate(function() {*

*document.querySelector("input[name='u']").value = "kimbang987";*

*document.querySelector("input[name='p']").value = "bang123";*

*document.querySelector("#frmLogin").submit();*

*console.log("Login submitted!");*

*});*

*page.render('login\_zing\_s2.png');*

*window.setTimeout(function () {*

*page.render('login\_zing\_s3.png');*

*phantom.exit();*

*}, 5000);*

*}*

*});*

### Biên dịch :*$ phantomjs $zing.js*

### $: đường dẫn

### Kết quả:

### Chương trình tải trang web zing.vn rồi thực hiện đăng nhập vào Website với tài khoản như trên mã nguồn.

### Kết quả được hiển thị ra ba bức ảnh: 'login\_zing\_s1.png', 'login\_zing\_s2.png' và 'login\_zing\_s3.png'.

### Đây là một cách duyệt Web, bởi nó cũng tải Website trên server về và thao tác như các trình duyệt Web thông thường.

### weather.js

### Mã nguồn có thể tải về trên trang chủ của PhantomJS.

### Khi biên dịch chương trình, kết quả sẽ là thông tin thời tiết cập nhật tức thời của một địa điểm nào đó do người dùng thiết lập.

### Ví dụ thông tin thời tiết khi thành phố là *“Mountain View, California”:*

### 

### Khi thành phố là “Hà Nội”

### 

### Chương trình weather.js cho thấy tính năng truy xuất tức thời vào dữ liệu DOM, CSS của một Website.

# Ứng dụng thực tế của PhantomJS

1. **YSlow**
2. **YSlow là gì?**

YSlow là một tiện ích được gắn trên trình duyệt Web dùng để phân tích trang Web dựa trên chuẩn các quy tắc của Yahoo. Qua đó cải thiện hiệu suất của Website.

1. **YSlow và PhantomJS.**

YSlow cho PhantomJS là một kịch bản dòng lệnh cho phép phân tích hiệu suất trang Web trực tiếp từ URL.

YSlow cung cấp cho PhantomJS các chuẩn đầu ra mới, sử dụng kiểm tra tự động: TAP (Test Anything Protocol) và Junit. Các chuẩn đầu ra mới vẫn đang được phát triển.

1. **Ví dụ**

Sau đây là ví dụ của YSlow cho PhantomJS.

***Cài đặt:***

* Cài đặt PhantomJS
* Vào trang chủ của YSLow (yslow.org) tải về file *yslow.js*

***Hỗ trợ:***

Gõ lệnh: ($: đường dẫn đến thư mục yslow.js)

*$ phantomjs yslow.js - help*

Kết quả sẽ hiển thị ra các cú pháp hỗ trợ người dùng:

*Usage: phantomjs [phantomjs options] yslow.js [yslow options] [url ...]*

*PhantomJS Options:*

*http://y.ahoo.it/phantomjs/options*

*YSlow Options:*

*-h, --help output usage information*

*-V, --version output the version number*

*-i, --info <info> specify the information to display/log (basic|grade|stats|comps|all) [all]*

*-f, --format <format> specify the output results format (json|xml|plain|tap|junit) [json]*

*-r, --ruleset <ruleset> specify the YSlow performance ruleset to be used (ydefault|yslow1|yblog) [ydefault]*

*-b, --beacon <url> specify an URL to log the results*

*-d, --dict include dictionary of results fields*

*v, --verbose output beacon response information*

*-t, --threshold <score> for test formats, the threshold to test scores ([0- 100]|[A-F]|{JSON}) [80]*

*e.g.: -t B or -t 75 or -t '{"overall": "B", "ycdn": "F", "yexpires": 85}'*

*-u, --ua "<user agent>" specify the user agent string sent to server when the page requests resources*

*-vp, --viewport <WxH> specify page viewport size WxY, where W = width and H = height [400x300]*

*-ch, --headers <JSON> specify custom request headers, e.g.: -ch '{"Cookie": "foo=bar"}'*

*-c, --console <level> output page console messages (0: none, 1: message, 2: message + line + source) [0]*

*--cdns "<list>" specify comma separated list of additional CDNs*

*Examples:*

*phantomjs yslow.js http://yslow.org*

*phantomjs yslow.js -i grade -f xml www.yahoo.com www.cnn.com www.nytimes.com*

***Ví dụ các tiện ích cơ bản:***

* Thông tin cơ bản của JSON của Website *www.yahoo.com*

Câu lệnh:

*$ phantomjs yslow.js --info basic http://www.yahoo.com*

Kết quả:

*{"w":491065,"o":95,"u":"http%3A%2F%2Fwww.yahoo.com","r":60,"i":"ydefault","lt":966}*

* Thông tin cơ bản của trang Web *www.cnn.com*

*$ phantomjs yslow.js --info basic --format plain http://www.cnn.com*

Kết quả:

*size: 561.4K (561458 bytes)*

*overall score: D (64)*

*url: http://www.cnn.com/*

*# of requests: 137*

*ruleset: ydefault*

*page load time: 2576*

* Xác định CDN

*$ phantomjs yslow.js -i grade --cdns "foo.com,bar.com,baz.com" example.com*

* Kiểm tra TAP cơ bản

*$ phantomjs yslow.js --info basic --format tap --threshold B http://yslow.org*

Kết quả:

*TAP version 13*

*1..1*

*ok 1 B (86) overall score*

* Kiểm tra TAP chi tiết.

*$ phantomjs yslow.js --info grade --format tap --threshold '{"overall": "B", "ycdn"*

*: 65}' http://yslow.org*

Kết quả:

*TAP version 13*

*1..24*

*ok 1 B (88) overall score*

*not ok 2 C (72) ynumreq: Make fewer HTTP requests*

*---*

*message: This page has 7 external Javascript scripts. Try combining them into one.*

*This page has 5 external stylesheets. Try combining them into one.*

*...*

*ok 3 C (70) ycdn: Use a Content Delivery Network (CDN)*

*---*

*message: There are 3 static components that are not on CDN. <p>You can specify CDN hostnames in your preferences. See <a href="https://yslow.org/faq#faq\_cdn">YSlow FAQ</a> for details.</p>*

*offenders:*

*- "yslow.org: 1 component, 8.0K (8.0K GZip)"*

*- "fonts.googleapis.com: 1 component, 1.0K (1.0K GZip)"*

*- "widgets.twimg.com: 1 component, 0.8K"*

*...*

*ok 4 A (100) yemptysrc: Avoid empty src or href*

*not ok 5 F (12) yexpires: Add Expires headers*

*---*

*message: There are 8 static components without a far-future expiration date.*

*offenders:*

*- "http://yslow.org/stylesheets/styles-min.css"*

*- "https://fonts.googleapis.com/css?family=Lato:300italic,700italic,300,700"*

*- "http://widgets.twimg.com/j/2/widget.css"*

*- "http://www.google-analytics.com/ga.js"*

*- "http://widgets.twimg.com/j/2/widget.js"*

*- "http://d.yimg.com/jc/ydn/yslow-ss.png"*

*- "http://widgets.twimg.com/i/widget-logo.png"*

*- "http://yslow.org/favicon.ico"*

*...*

*ok 6 A (100) ycompress: Compress components with gzip*

*ok 7 A (100) ycsstop: Put CSS at top*

1. **Qunit**

Vào trang chủ của PhantomJS tải hai file: run-qunit.js và test.html

Biên dịch file trên như sau:

$phantomjs run-qunit file://`pwd`/test.html

Kết quả chương trình:

*$ phantomjs run-qunit.js file://`pwd`/test.html*

*'waitFor()' finished in 200ms.*

*Tests completed in 21 milliseconds.*

*5 tests of 5 passed, 0 failed.*

Giải thích kết quả:

* *'waitFor()' finished in 200ms:* thời gian chạy chương trình là 200ms
* Tiếp theo ý nghĩa là số kiểm thử được hoàn thành

Kết luận: Qunit cho PhantomJS giúp người dùng thực hiện kiểm thử thời gian chạy và tính chính xác của một sự kiện trên Web.

# Tổng kết

Trên đây là bản báo cáo về PhantomJS.

Bản báo cáo vẫn còn khá sơ sài, đặc biệt về phần Ứng dụng thực tế. Vì thế

nhóm sẽ tiếp tục tìm hiểu về phần này và bổ sung vào bản cáo cuối cùng.

**Tài liệu tham khảo**

<http://phantomjs.org/> // Trang chủ phantomJS

<http://ariya.ofilabs.com/about> // Thông tin nhà phát triển

<http://phantomjs.org/release-names.html> // Các phiên bản của phantomJS

<http://yslow.org/> // Yslow

<https://gist.github.com/gmarik/1305062> // Qunit

-HẾT-